Amendments to the Specification

Replace paragraph [0011] with the following:

Data integrity is a major factor in drive development. An accurate measurement for unrecoverable error events allows the disc drive designer to optimize encoding schemes and ECC algorithms for the disc drive ~~recoding~~ recording medium. There is therefore a significant need for improvements in the art to enable a disc drive designer to readily and accurately characterize different alternative configurations of a digital data channel, including the use of different RLL and ECC encoding schemes, symbol lengths and interleaves.

Replace paragraph [0055] with the following:

FIG. 7 generally represents a readback circuit 165 used to recover the data encoded by FIG. 6. A retrieved bit stream on path 166 is separated into the RLL encoded user data and the RLL encoded code words, the latter of which are decoded by a {fraction (20/21)} RLL decoder block 168. The decoded code words are then used for ECC detection and correction of the RLL encoded user data by an ECC decoder 170. If successful, corrected RLL encoded user data are output on path 172 to a second RLL ~~encoder~~ decoder block 174 which applies {fraction (99/100)} decoding to output the original user data on path 176.

Replace paragraph [0070] with the following:

It will be contemplated that the digital channel 230 of FIG. 10 is configured such that data supplied as an input to a selected disabled block (such as the RLL encoder 246) passes through the disabled block without effect or change in the data. In this way, the system 200 can use a single path (266) to introduce write data at various points in the write channel 234 and can use a single path (NRZ path 262) to obtain readback data from various points ~~along~~ in the read channel 236.

Replace paragraph [0071] with the following:

At step 306, selected test input data are supplied by the host computer 202 to the disc drive 100. The input data are preferably stored in a first memory location of the RAM 206 during this step. The system proceeds in this example in step 308 to encode the input data with the selected RLL encoding (such as {fraction (8/9)}). No ECC encoding of the data (either before or after the RLL encoding) takes place in this example. At step 310, the RLL encoded data are written to the appropriate data sectors 122 on the discs 108 (media).

Replace paragraph [0073] with the following:

At step 316, the system 200 arranges the retrieved data into corresponding symbols and interleaves and determines the number of uncorrectable symbols during step 318. These steps are performed for each of a number of different symbol ~~length~~ lengths in turn, as indicated by decision step 320.

Replace paragraph [0083] with the following:

Generally, CBER is a measure of an expected error rate among corrected bits (i.e., with the operation of the ECC scheme) and is a better indicator of actual performance of a particular channel configuration as compared to RBER. CBER takes into account the performance of the ECC as it relates to the encoding scheme. Since CBER is an end to end ~~recoding~~ recording system measurement, the disc drive designer can optimize the density gain of the ~~recoding~~ recording medium for each specific product platform. It is not necessary to guess what the end performance will be as with the RBER. It will be noted that the CBER value can be determined in other ways (such as on an actual number of erroneous bits basis or on a per total symbols read basis), but such variations will be proportional to the value set forth by equation (2) and so the particular form is up to the preferences of a particular user.

Replace paragraph [0095] with the following:

The method preferably comprises steps including using a digital data channel to store input data to a recordable medium (such as 108), said input data comprising an input stream of data bits (such as by step 310); subsequently using the digital data channel to obtain readback data from the recordable medium, said readback data comprising an output stream of data bits corresponding to the input stream of data bits (such as by step 312); arranging the input data into an input sequence of multibit symbols, each symbol having a first selected symbol length (such as by step ~~316~~ 308); arranging the readback data into an output sequence of multibit symbols, each symbol having the first selected symbol length (such as by step 316); and comparing the output sequence with the input sequence to determine a first number of erroneous symbols in the output sequence (such as by step 318).

Replace paragraph [0099] with the following:

The method further preferably comprises additional steps of characterizing the input sequence and the output sequence as a first input sequence and a first output sequence (such as by step 320); further arranging the input data into a second input sequence of multibit symbols, each symbol having a second selected symbol length different from the first selected symbol length (such as by step ~~316~~ 308); further arranging the readback data into a second output sequence of multibit symbols, each symbol having the second selected symbol length (such as by step 316); and comparing the second output sequence with the second input sequence to determine a second number of erroneous symbols in the second output sequence (such as by step 318).